ZJU Summer 2024 Contest 5

Contest Analysis

Group C

07.10.2024

A. Area in Circle

Description

在一个定圆中，给定一些点。加一个也在圆内的点，使得最终凸包面积最大化。

A. Area in Circle

Solution

首先，显然新加的点应该在圆周上。如果不在圆周上，将这个点向着圆心反方向移动到圆周，能使面积更大。
新加入的点会成为凸包上的一个顶点，这个顶点在凸包上会有相邻的两个点（称为前驱点和后继点）。
对原凸包的边作延长线交圆周，交点将圆周分割为不超过 $2n$ 个弧。可以发现，这里每一段弧都对应着唯一的一对前驱后继点。

A. Area in Circle

Solution

考虑顺序枚举弧，同时维护弧上的节点对应的前驱、后继点（记为 U, V）及原凸包上这一部分的面积。找到弧上离直线 UV 最远的点，就是加入的点。可以通过几何方法或者三分求出。
时间复杂度 $O(n \log n)$

B. Cyclic Strings

Description

给出字符串 S，支持两种操作：
- 将 S 循环左移一位
- 求本质不同第 x 小子串

B. Cyclic Strings

Solution

用 suf_i 表示以**一开始的串**的第 i 个下标为开头的后缀
做循环的 SA，求得 sa_i 表示 rank 为 i 的后缀是 suf_{sa_i}，若两个后缀在循环意义下相等，则按照下标顺序。（也可以倍长以后求 SA ）
计算 L_i，表示 suf_i 在长度超过 L_i 时不作为 SA 中其他串的前缀出现

L_i 的计算：
记

$$D_i(j) = \begin{cases} i-j, & i \geq j \\ n+i-j, & i < j \end{cases}$$

表示当 suf_j 比 suf_i 更长时，比 suf_i 长多少
则当 $|suf_i| \leq n - D_i(j)$ 时，suf_j 比 suf_i 更长。
则

$$L_i = \max_j \min\{LCP(suf_i, suf_{sa_j}), n - D_i(sa_j)\}$$

记

$$u_j = LCP(suf_i, suf_{sa_j})$$

$$v_j = \begin{cases} \max_{k=j}^{rank_i - 1} n - D_i(sa_k), & j < rank_i \\ \max_{k=rank_i+1}^{j} n - D_i(sa_k), & j > rank_i \end{cases}$$

考虑由于 $LCP(suf_i, suf_{sa_j})$ 随 j 远离 $rank_i$ 递减，则用 v_j 代替 $n - D_i(sa_j)$ 不会改变答案，则上式实际上等价于

$$L_i = \max_j \min\{u_j, v_j\}$$

B. Cyclic Strings

Solution

这里随着 j 远离 $rank_i$，u_j 递减，v_j 递增，可以考虑求得二者大小关系变化的分界点来求得答案。
由于 u_j 可以通过预处理 $O(1)$ 快速求得，考虑用线段树维护 $n - D_i(sa_j)$ 的值，v_j 为一个区间 max，然后在左右两侧做线段树上二分。
这个线段树对于不同的 i 是不同的，但对于 $i = t$ 和 $i = t+1$ 的情况，二者的线段树相差一个元素。所以可以按照在原串中下标的顺序进行计算。
于是我们可以用 $O(n \log n)$ 的复杂度求出 L_i。

B. Cyclic Strings

Solution

然后，我们只保留不是 SA 上前面某个串的前缀的串，维护它们的长度和 *height*

对于循环左移的操作，即是从这些串中删除一个，然后所有串长度 $+1$，之后加入所有长度变为 $L_i + 1$ 的串。

在串长 $+1$ 时，由于可以保证被加入的串不存在前缀关系，*height* 不变。

加入/删除串时，新的 *height* 可以 $O(1)$ 求出。用线段树维护这些串，n 次循环左移操作的总时间复杂度为 $O(n \log n)$，即均摊每次循环左移 $O(\log n)$

询问是一个线段树上二分。

总时间复杂度 $O((n + Q) \log n)$

C. Extra Shot

Description

进行 n 轮攻击，每一轮会先普通攻击一次，然后有固定概率触发 m 种额外攻击的一种，第 i 种触发概率为 p_i。
如果额外攻击击中敌人，那么会再进行一次普通攻击，这次普通攻击同样有可能触发额外攻击。但是如果触发了和上一次一样的额外攻击，这次额外攻击会 miss 并且结束这一轮攻击。
问最终期望有多少次击中的额外攻击。

C. Extra Shot

Solution

记 f_i 表示触发第 i 种额外攻击以后，期望还命中多少次额外攻击。

$$f_i = 1 + \sum_{j \neq i} p_j f_j$$

设 $g_i = p_i f_i$，$S = \sum_{i=1}^{n} g_i$
则

$$g_i = p_i(1 + S - g_i)$$

C. Extra Shot

Solution

整理得 $g_i = \frac{p_i(1+S)}{1+p_i}$

将这个东西代入 S 的定义，可以解得 S 的值。答案为 nS

D. Jenga

Description

一个序列 a $(|a_i| \leq n)$，每次可以选择一个下标 i，对所有 $j > i$，令 $a_j+ = a_i$

在 $3n$ 次操作内将序列变为非严格单调的。

D. Jenga

Solution

考虑第一个不为零的数 a_j。
若 $a_j > 0$，考虑贪心将所有数修改为正数，逆序遍历，每遇到 $a_i < a_{i-1}$，进行一次操作 $i-1$。
负数同理。
操作次数不超过 $3n$（实际上是 $2n$）

E. Perfect Square

Description

定义一个集合是好的，如果其中任意两个数的乘积是完全平方数。
对于一个集合，单次修改可以选择其中一个数乘上一个质数，一个集合的价值为将集合变成好的集合的最少操作次数。
给定 n 个数和 q 个询问，每次询问一个区间的价值。
强制在线。

E. Perfect Square

Solution

集合是好的，即每个质因子在集合中每个数的幂次同奇偶。
因此集合 S 的价值为 $\sum_p \min(c_p, |S| - c_p)$ ，其中 c_p 为集合中 p 的幂次为奇数的数的个数。
将其转化为 $\sum_p c_p + \sum_{c_p > |s|/2} |s| - 2c_p$。其中 $\sum_p c_p$ 可以使用前缀和求得。

E. Perfect Square

Solution

令 w 为 1e7 内整数包含不同质因子数量的最大值（小于 10）。
令 $len = \sqrt{nw}$，对于区间长度小于 len 的询问，直接暴力求解，可以使用 unordered_map 来对 c_p 进行计数，复杂度为 $O(w\sqrt{nw})$。

E. Perfect Square

Solution

因为对于区间 $[1, n]$，$c_p^{[1,n]}$ 大于 $len/2$ 的质数最多有 $nw/len * 2 = O(\sqrt{nw})$ 个，可以提前预处理保存下这些质数，并且对这些质数分别开一个前缀和数组来计数。
对于区间长度大于 len 的询问，只需要检查上述质数是否满足 $c_p > |S|/2$ 并计算贡献即可。
最终复杂度为 $O(qw\sqrt{nw})$。

F. Rainy Day

Description

有 n 根柱子，将柱子任意排列，如果一根柱子左右两边都有比它高的柱子，它上面就会积水，水量为 $h_m - h_i$，h_m 为两侧最高的柱子中的较小高度，h_i 为当前柱子高度。
求可能的柱子排列中，所有出现的积水总量。

F. Rainy Day

Solution

首先可以直接将最高的柱子直接放在最右边，显然这不会遗漏任何情况。
在此基础上，我们先将柱子排序，然后从小到大依次考虑每个柱子。对于第 i 个柱子 $h_i (1 \leq i < n)$，我们可以将其插入到第 $j (i < j < n)$ 个柱子之后，并获得 $h_j - h_i$ 的贡献。
而考虑到第 i 个柱子时，如果这个柱子在此前被某根柱子插入，我们可以转换一下，将插入的柱子代替第 i 个柱子，这样就可以复用先前的考虑，变成一个分组背包的模型，复杂度 $\Theta(n^3 h)$。

F. Rainy Day

Solution

核心思想有了之后，有很多种优化方式。
例如，我们可以使用 bitset 优化到 $O(\frac{n^3 h}{\omega})$；
可以转化一下模型模型，转移只从 i 到 $i+1$，考虑第 i 个物品时记录体积为 k 的情况下最多有多少个物品，再用单调性优化到 $O(n^2 h)$；
也可以用生成函数加速背包，优化到 $O(nh \log nh)$

Thanks!